

Chapter 2

Bioinformatics Approaches and Software for Detection of Secondary Metabolic Gene Clusters

Natalie D. Fedorova, Venkatesh Muktali, and Marnix H. Medema

Abstract

The accelerating pace of microbial genomics is sparking a renaissance in the field of natural products research. Researchers can now get a preview of the organism's secondary metabolome by analyzing its genomic sequence. Combined with other -omics data, this approach may provide a cost-effective alternative to industrial high-throughput screening in drug discovery. In the last few years, several computational tools have been developed to facilitate this process by identifying genes involved in secondary metabolite biosynthesis in bacterial and fungal genomes. Here, we review seven software programs that are available for this purpose, with an emphasis on *antibiotics & Secondary Metabolite Analysis SHell* (antiSMASH) and *Secondary Metabolite Unknown Regions Finder* (SMURF), the only tools that can comprehensively detect complete secondary metabolite biosynthesis gene clusters. We also discuss five related software packages—*CLUster SEquence ANalyzer* (CLUSEAN), *ClustScan*, *Structure Based Sequence Analysis of Polyketide Synthases* (SBSPKS), *NRPS Predictor*, and *Natural Product searcher* (NP.searcher)—that identify secondary metabolite backbone biosynthesis genes. This chapter offers detailed protocols, suggestions, and caveats to assist researchers in using these tools most effectively.

Key words: Fungi, Genome, Gene cluster, Secondary metabolite, Mycotoxin, Polyketide, Nonribosomal peptide, Natural product, Antibiotic, Software

1. Introduction

Secondary metabolite (SM) biosynthesis pathways represent the main source of metabolic diversity in many bacterial and fungal species and have been a gold mine of new antibiotics and pharmaceuticals for decades. Although SMs have been described in plants, green algae, and protists, fungi and bacteria produce the vast majority of these natural compounds. Thus, a single bacterial or fungal genome can encode biosynthetic enzymes for over 60 SMs, such as polyketides, nonribosomal peptides, isoprenoids, flavonoids,

and many others. In these organisms, genes involved in secondary metabolite biosynthesis are often co-regulated and physically linked into gene clusters on the chromosomes. This salient feature, along with sequence conservation, aids in computational discovery of new SM biosynthetic pathways encoded by microbial genomes (1, 2). Fueled by the torrent of genomic sequences and other -omics data, *in silico* approaches hold much promise for cost-effective alternatives to industrial high-throughput screening in drug discovery.

To date, several computational tools have been developed for detection of SM biosynthetic pathways in sequenced fungal or bacterial genomes. In addition to SM gene identification, some tools can predict substrate specificity of polymerization enzymes, generate approximations for products' 3D structures, and more. The usability of the tools, however, varies greatly depending on the quality of the sequence data, the user's needs, and his/her computer proficiency. Even though detailed manuals are available, funding limitations often prevent "freeware" developers from creating a perfect user-friendly interface. To assist researchers in using the software efficiently, this chapter describes approaches, software packages, and step-by-step protocols for the use of these tools. The main focus is on *Secondary Metabolite Unknown Regions Finder* (SMURF) and *antibiotics & Secondary Metabolite Analysis SHell* (antiSMASH), the only software packages that can detect the entire SM gene clusters in microbial genomes (3, 4). We also discuss five more specialized software tools (Table 1): CLUster SEquence ANalyzer (CLUSEAN), NRPSPredictor, ClustScan, Structure Based Sequence Analysis of Polyketide Synthases (SBSPKS), and Natural Product *searcher* (NP.searcher) (5–10).

2. Materials

2.1. Operating System and Hardware Requirements

Except for CLUSEAN, most tools described here (antiSMASH, SMURF, ClustScan, SBSPKS, NRPSPredictor, and NP.searcher) are platform independent and work equally well on Windows, Mac, or Linux environments and are available as Web servers (Table 1). These simply require a computer with an Internet connection and a Web browser like Mozilla Firefox, Google Chrome, Apple Safari, or Microsoft Internet Explorer. antiSMASH also has a stand-alone program available, for which a computer with at least 1.0 GHz CPU and 2 GB of RAM memory is recommended, as well as a Windows XP/Vista/7, Linux Ubuntu, or Mac OS X 10.6+ operating system. Except for ClustScan, all tools are freely available and accessible to users without any programming skills.

Table 1
Software tools for detection of SM genes and gene clusters

Tools	System requirements	Input	Output	Reference
antiSMASH	<ul style="list-style-type: none">– Web server: Any recent Web browser in any OS– Stand-alone version: Windows XP/Vista/7, Ubuntu Linux, or Mac OS X 10.6+	<ul style="list-style-type: none">– List of GenBank/RefSeq accession numbers OR– DNA sequences in GenBank/EMBL/FASTA formats– Bacterial and fungal sequences	<ul style="list-style-type: none">– Interactive output with annotation and structure for each cluster– Graphical representation– Homologous clusters– BLAST/Pfam results– Functional predictions– Backbone genes and clusters	(4)
SMURF	Any Web browser in any OS	<ul style="list-style-type: none">– Protein sequences in FASTA format AND– Tab-delimited text file containing chromosomal coordinates of genes– Fungal sequences	<ul style="list-style-type: none">– List of predicted backbone genes– Functional predictions for genes– List of predicted clusters– Backbone genes and clusters	(3)
CLUSEAN	<ul style="list-style-type: none">– Preferred OS: Linux/Unix; except for NRPSPredictor– Compatible with MS Win 2000, XP, and VISTA– Installation required:– NCBI BLAST– HMMer– SVMlight– NRPSPredictor– Artemis Perl 5.8 or higher	<ul style="list-style-type: none">– EMBL-formatted DNA files– Bacterial and fungal sequences	<ul style="list-style-type: none">– Domain content– Functional predictions for genes– Substrate specificity predictions– NRPS and PKS genes only	(5)

(continued)

Table 1
(continued)

Tools	System requirements	Input	Output	Reference
ClustScan	Java Runtime Environment	<ul style="list-style-type: none"> – DNA sequences in FASTA or any format supported by ReadSeq – Bacterial and fungal sequences 	<ul style="list-style-type: none"> – Annotated genes – Domain prediction – Browsing by domain searches – Backbone genes only 	(6)
SBSPKS	Any Web browser in any OS	<ul style="list-style-type: none"> – Protein sequences in FASTA format (<10) – Sequences have to be entered manually by copying and pasting – Bacterial and fungal sequences 	<ul style="list-style-type: none"> – Visualization of PKS domains – Prediction of docking order of PKSs – Prediction of PKS 3D structures – PKS genes only 	(7)
NRPSPredictor	Any Web browser in any OS environment	<ul style="list-style-type: none"> – Protein sequences in FASTA format OR – Extracted domain A signatures – Bacterial and fungal sequences 	<ul style="list-style-type: none"> – List of domains – NRPS substrate specificity – SVM scores for domains – NRPS and NRPS-PKS genes only 	(9)
NP.searcher	Any Web browser in any OS	DNA sequences in FASTA format <ul style="list-style-type: none"> – Bacterial and fungal sequences 	<ul style="list-style-type: none"> – Coordinates for NRPS/PKS genes – Number of PKS/NRPS modules – Backbone genes only 	(8)

2.2. Input Sequence Data Formats

Correctly formatting input files is critical for getting meaningful results from any software tool. Yet for a biologist with no computational background, this step can sometimes present a serious obstacle. Also, input sequence data formats may vary depending on the software used. Below, we provide a few examples of data formats used by the tools for detection of SM genes (Fig. 1).

FASTA files. The FASTA format is a standard text-based format used in bioinformatics to represent nucleotide or peptide sequences. These sequences are usually preceded by sequence identifiers (IDs), names, and comments, collectively referred as FASTA headers. Most algorithms described here accept the NCBI FASTA specification (<http://blast.ncbi.nlm.nih.gov/blasts.cgi?help.shtml>). Files containing more than one nucleotide or protein sequence are sometimes called multiFASTA files. SMURF for example takes multiFASTA files that contain protein IDs in FASTA headers and sequences (Table 1).

GenBank and EMBL files. Both GenBank flat file and EMBL formats (Fig. 1) are rich formats used for storing nucleotide and protein sequences and meta-information. They are also the standard formats accepted by the International Nucleotide Sequence Database Collaboration, which includes the DNA DataBank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank at NCBI. The two formats are highly similar and contain common meta-data fields such as the accession number, source organism, literature references, as well as key genomic features (protein-coding genes, RNAs, exons, introns, etc.) and their coordinates. Detailed information about these formats is available at <http://www.ebi.ac.uk/help/formats.html>.

Tab-delimited TXT files. This format was designed to store data by separating the values in each row with a tab. Alternative types of delimited TXT files have other specific delimiter characters such as the comma, colon, space, and vertical bar (pipe). The lines are separated by line breaks. For example, SMURF accepts tab-separated files where data items are separated by the tab (Fig. 1). Tab-delimited TXT files can be easily generated using a spreadsheet program such as Microsoft Excel, by using the “Save as TXT file” option from the top drop-down File menu.

3. Methods

3.1. Choosing the Right Approaches and Software for the Job

Several software tools are currently available for identification of SM biosynthesis genes and gene clusters in bacterial fungal genomes (Table 1). Most tools approach the problem of SM gene detection using the LEGO principle, which states that most biological complexity is generated by using only a small number of building

FASTA Format

```
>gi|211587300|emb|AM920435.1| Penicillium chrysogenum Wisconsin 54-1255
complete genome, contig Pc00c20
TCTTCAACATATGATGTTTCCCGATGATCCATTCGGTCTGTACATCCCCCAGACGCCAAACGAATATTC
AACTTTGGCTTACCATGAAGATAATCTGGGAAAGCGATTTCACAAGAACAAGAACTACAATTGATGAAG
TTTGATGGGCTCCACAAAATAGTTCTATACATTACTAAACATTACAACCTCTCTTACTAGCACAGAAAAC
TTTCAATACGCATTTTAGAGAAAGCCACGGTTAATCCAATTGAATTAACGTTGTATCTTCGTCATCAGCG
TTTATTACCGCGTGTATCTCATTAGTGCA
...

```

GenBank Format

```
LOCUS       AM920435             3652229 bp      DNA      linear      PLN 01-NOV-2008
DEFINITION  Penicillium chrysogenum Wisconsin 54-1255 complete genome
ACCESSION   AM920435
VERSION     AM920435.1  GI:211587300
SOURCE      Penicillium chrysogenum Wisconsin 54-1255
FEATURES             Location/Qualifiers
     source          1..3652229
                     /organism="Penicillium chrysogenum Wisconsin 54-1255"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:500485"
     gene            complement(<967..>2641)
                     /locus_tag="Pc20g00010"
     CDS             complement(join(967..1004,1054..1224,1273..2107,
                     2138..2641))
                     /locus_tag="Pc20g00010"
                     /protein_id="CAP85330.1"
...

```

EMBL Format

```
ID      AM920435; SV 1; linear; genomic DNA; STD; FUN; 3652229 BP.
XX
AC      AM920435;
XX
DE      Penicillium chrysogenum Wisconsin 54-1255 complete genome
XX
FH      Key          Location/Qualifiers
FH
FT      source          1..3652229
FT                      /organism="Penicillium chrysogenum Wisconsin 54-1255"
FT                      /mol_type="genomic DNA"
FT                      /db_xref="taxon:500485"
FT      CDS             complement(join(967..1004,1054..1224,1273..2107,
FT                      2138..2641))
FT                      /gene="Pc20g00010"
FT                      /locus_tag="Pc20g00010"
...

```

Fig. 1. FASTA, GenBank, and EMBL data formats. All software packages described here accept input files in at least one of these formats.

blocks. Indeed, analysis of characterized SM pathways shows that most biosynthetic reactions are performed by a limited number of enzyme classes, which are responsible for the enormous spectrum of natural products produced by bacteria and fungi (11). Consequently, known SM enzymes share highly conserved protein domains, which can be identified by sequence similarity.

Additionally, identification of new pathways is greatly facilitated by SM gene clustering and the presence of the so-called backbone genes, which are easy to detect based on their domain content. They encode enzymes that catalyze the polymerization reaction, which generates the product's backbone and is often the first reaction in an SM pathway. Backbone genes are typically surrounded by genes encoding "decorating enzymes," which catalyze subsequent reactions such as oxidation, reduction, methylation, and glycosylation of the backbone molecule. These reactions may alter the compound's hydrophilicity, stability, subcellular localization, and bioactivity. The end product is often translocated outside the cell by transporters, also encoded within a typical SM cluster. Finally, clusters may contain additional genes encoding precursor biosynthesis enzymes, additional transporters, detoxification enzymes, or pathway-specific transcriptional regulators. Most of these proteins contain one or more SM-related domains, which can be identified computationally.

Since most available tools use similar approaches, the choice of the software largely depends on the desired output, quality of the sequence data, and taxonomy of the organism of interest (Table 1). All seven software tools use protein domain searches to reliably predict backbone genes. SMURF and antiSMASH are the only tools that examine neighboring regions on the chromosome to predict the entire gene clusters. SMURF is specific to fungal genomes, and generates a quick report as a single complete and easily parsable TXT file. antiSMASH identifies clusters in both bacterial and fungal genomes. It also generates an interactive graphical output with integrated additional analyses and cross-links to other services. For analysis of fungal genomes, it is a good idea to use both tools because they rely on different algorithms for predicting cluster boundaries (see Note 2).

3.2. antibiotics and Secondary Metabolite Analysis Shell

antiSMASH is a comprehensive pipeline for identification of gene clusters encoding biosynthetic enzymes for a wide range of SM compound classes (4). It identifies polyketides, nonribosomal peptides, terpenes, aminoglycosides, aminocoumarins, indolocarbazoles, lantibiotics, bacteriocins, nucleosides, beta-lactams, butyrolactones, and others in both fungal and bacterial nucleotide sequences. The underlying algorithm is presented in Fig. 2. The antiSMASH software is available at <http://antismash.secondarymetabolites.org>, both as a Web server and as a stand-alone application.

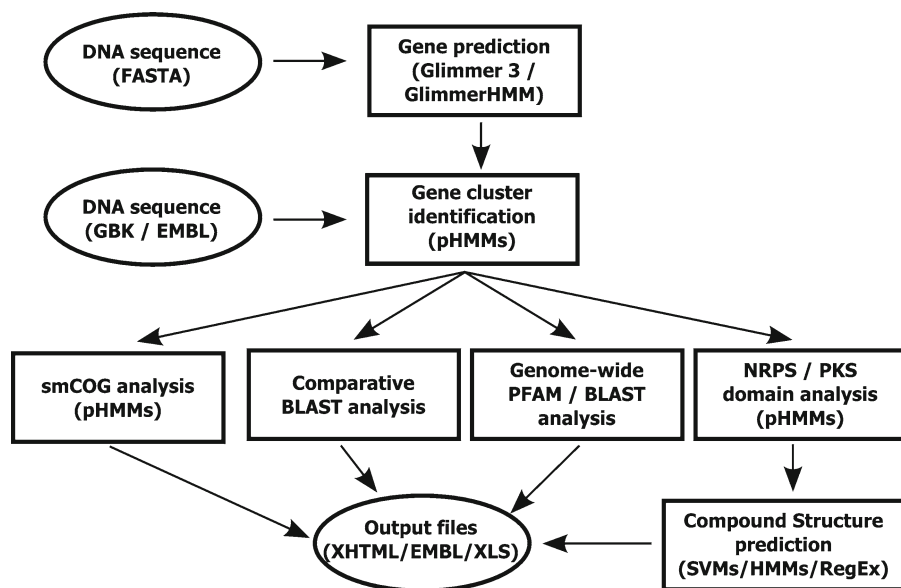


Fig. 2. Key steps in the antiSMASH algorithm. The figure shows principle steps in the algorithm from a DNA sequence to an interactive output file. Adapted from (4).

3.2.1. Preparing Input Data

Besides accepting nucleotide accession numbers from GenBank or RefSeq, antiSMASH accepts fully annotated GenBank and EMBL files as well as FASTA files that contain a single DNA sequence. MultiFASTA files are currently not accepted and should be split into single-sequence FASTA files. When preparing input files manually, it is critical to use a plain text editor like Notepad or Emacs and to save the files as “All files (*.*)” with the correct file extension (.fasta, .fas, or .fa for FASTA files; .gbk or .gb for GenBank files; .embl or .emb for EMBL files).

For a wide range of fungal and bacterial species, an annotated genome sequence is already available at NCBI. In such case, a corresponding genome accession number can simply be inserted into the third text box on the antiSMASH Web site, labeled as “input GenBank/RefSeq accession number of desired file”. Accession numbers can be acquired by querying the NCBI Entrez system (<http://www.ncbi.nlm.nih.gov/>). The search can be performed by selecting the Bioproject (Genome Project) database from the top pull-down menu and entering the organism name in the search box as a query (e.g., “*Aspergillus fumigatus* AF293 [Organism]”). If more than one genome build is available for a particular organism, it is recommended to use the most current build. Accession numbers of any nucleotide GenBank/RefSeq entries that are smaller than full genomes can also be used.

If no genome accession number is available for the species of interest, its annotated DNA sequence can be entered as an EMBL

or GBK file in the second text box (labeled “upload a file”) in the antiSMASH front end. These files can be generated from a FASTA sequence file and annotation table as described below:

1. Download and install Python from <http://www.python.org/download/>.
2. Download the GenBank formatting script from the antiSMASH Web site <http://antismash.secondarymetabolites.org/downloads.html>.
3. In a spreadsheet program such as Microsoft Excel, prepare a table with the coordinate information on each exon/gene structure in the following columns: (1) name of contig FASTA file, (2) gene locus tag (must be unique), (3) 5' exon/gene start, (4) 3' exon/gene end, (5) gene annotation. For every exon or gene, fill in the information in a single line. Make sure that all exons from one gene have the same locus tag and that the exons/genes are listed in the same order in which they occur on the DNA sequence. Save the table as a tab-delimited TXT file “annotationtable.txt”.
4. Copy ‘annotationtable.txt’ and the config FASTA files to the same folder where the EMBL-formatting script is located.
5. Open the Terminal/Command Prompt, and navigate to the folder, which contains the files.
6. Type in “python format_embl.py” and press enter.
7. For every config, an EMBL file will be created that can serve as an input for antiSMASH.

Finally, if only an unannotated DNA sequence is available, a FASTA file with the unannotated nucleotide sequence can be uploaded to antiSMASH. The algorithm will then run a gene prediction algorithm: GlimmerHMM (12) for eukaryotic sequences and Glimmer3 (13) for prokaryotic sequences to perform a gene annotation of the DNA sequence before running the rest of the pipeline (see Note 3). Alternatively, the genomic sequence can be annotated using standard genome annotation tools, converted into GenBank or EMBL format, and then submitted to antiSMASH. Automated genome annotation can be performed using automated Web pipelines such as the MAKER Web Annotation Service (14) or RAST (for prokaryotic sequences only) (15).

3.2.2. Choosing Parameters and Running antiSMASH

One of the main advantages of using antiSMASH is that it allows users to run other analyses (in addition to gene cluster identification) by selecting various options at the front end (Fig. 2). The first option allows user to select cluster types such as polyketides, nonribosomal peptides, terpenes, or aminoglycosides. Subsequently, the user can choose whether or not to include a number of additional, computationally intensive analyses such as the following:

- Secondary metabolite Cluster of Orthologous Groups (smCOG) analysis: This module attempts to assign all genes to one of the smCOGs, which represent SM-specific gene families. For every assigned gene, a phylogenetic tree is generated to show the relatedness of that gene to other genes in the smCOG family.
- Gene Cluster Blast: For each gene cluster, this module detects putative homologs in a global database of gene clusters from across the tree of life, by searching for homologous genes and synteny (conserved gene order) inferred from BLASTp hits among gene clusters.
- Whole genome BLAST and PFAM analyses: This module executes genome-wide analyses from the CLUSEAN pipeline. It provides PFAM domain annotations for each gene in the DNA sequence, as well as BLAST hits to a custom nonredundant database of bacterial and fungal genes. Based on these results, it also generates annotation suggestions for each gene. Finally, if the PFAM analysis option is selected, the module generates a graph that displays genomic regions enriched in SM-related PFAM domains.

To very quickly obtain gene cluster predictions for a genome, the user may choose not to include all the analyses available. However, choosing these additional options can be beneficial for more in-depth analysis of gene clusters. Domain architecture analysis of PKS and NRPS proteins is automatically included in each analysis and therefore does not need to be selected.

When an antiSMASH run is finished, it generates an interactive XHTML file containing the output. Depending on the options chosen, this file will include a functional description of each gene cluster, its domain annotations for PKS and NRPS genes, a list of homologous clusters in other species, a putative chemical structure of the cluster's product, and several downloadable output files. All sections have a question mark button at the top right to provide more information on each section. Additionally, the user can access more detailed information by clicking on elements of the page or hovering over them with the mouse. To store antiSMASH results locally, the user can click the "Download all results" button at the bottom right. Doing so will allow access to the antiSMASH results at any later point in time, while the output page on the server will be deleted eventually.

3.2.3. Viewing antiSMASH Results in a Genome Browser

For each run, antiSMASH generates an EMBL file that can be analyzed in a genome browser such as Artemis. This can especially be useful for viewing the whole-genome PFAM and BLAST analyses, which are only displayed in the EMBL file, but not in the general output XHTML file. To view the results in Artemis, follow these steps:

1. Download and extract the zip file that contains the results by clicking on the link “Download all results” at the bottom right of the antiSMASH output page.
2. Download and install Artemis from <http://www.sanger.ac.uk/resources/software/artemis/>.
3. Start Artemis, select “File » Open...”, and open the output file ending with “.final.embl” downloaded from antiSMASH.
4. The antiSMASH results should appear in the main Artemis window (Figs. 3 and 4). NRPS and PKS domains as well as conserved motifs within these domains are saved as “CDS_motif” or “misc_feature” features. More details for every feature can be viewed by selecting it and pressing Ctrl-V. If PFAM and BLAST options were selected, detected PFAM domains get saved as “CDS_motif” features. BLASTP results for each gene can be viewed by selecting a coding sequence (CDS) icon and pressing Ctrl-B (please note that the full download of all results is required for this option to be functional). The putative function of the CDS is displayed as a note.

3.2.4. Batch Processing of Multiple Files

Some users may want to analyze a large number of input files such as multiple genome sequences or a single draft genome that contains a large number of contigs. In such cases, the user can perform a batch run using the stand-alone version of antiSMASH in combination with a batch processing script, available from the Web site. To perform such a batch run, one needs to follow these steps:

1. Download the stand-alone version of antiSMASH for your operating system from <http://antismash.secondarymetabolites.org/download.html>.
2. Download and install Python from <http://www.python.org/download/>.
3. Download the script for batch processing from <http://antismash.secondarymetabolites.org/downloads.html> and copy it into the antiSMASH installation folder.
4. In the folder where antiSMASH is installed, create a subfolder “batch_input” and copy the files that need to be analyzed into this folder.
5. Alternatively, create a multiGBK/multiEMBL/multiFASTA file with your GBK/EMBL/FASTA entries.
6. Open the Terminal/Command Prompt, and navigate to the antiSMASH installation folder.
7. Type in a command with the structure “python batch_antismash.py <multiple_entry_file.gbk> --options” and press enter. The multiple entry file is optional. If not supplied, the script will search your ‘batch_input’ folder for single-entry files. You can add antiSMASH options to batch_antismash.py (e.g. “--clusterblast n --taxon e”), which will be forwarded to antiSMASH.)



antibiotics & Secondary Metabolite Analysis SHell

Nucleotide sequence input

- 1
- 1
- 2 Enter e-mail address here to receive results. (Optional)
- 3 Upload a file (GenBank / EMBL / FASTA formats accepted)
- 3 Or input GenBank/RefSeq accession number of desired file

- 4 ☐ DNA of Eukaryotic origin

Gene cluster types to search:

<input checked="" type="checkbox"/> all	<input checked="" type="checkbox"/> polyketides (type I)	<input checked="" type="checkbox"/> polyketides (type II)	<input checked="" type="checkbox"/> polyketides (type III)
<input checked="" type="checkbox"/> nonribosomal peptides	<input checked="" type="checkbox"/> terpenes	<input checked="" type="checkbox"/> lantibiotics	
5 <input checked="" type="checkbox"/> bacteriocins	<input checked="" type="checkbox"/> beta-lactams	<input checked="" type="checkbox"/> aminoglycosides / aminocyclitols	
<input checked="" type="checkbox"/> aminocoumarins	<input checked="" type="checkbox"/> siderophores	<input checked="" type="checkbox"/> ectoines	
<input checked="" type="checkbox"/> butyrolactones	<input checked="" type="checkbox"/> indoles	<input checked="" type="checkbox"/> nucleosides	
<input checked="" type="checkbox"/> phosphoglycolipids	<input checked="" type="checkbox"/> melanins	<input checked="" type="checkbox"/> others	

- 6 ☒ smCOG analysis for functional prediction and phylogenetic analysis of genes

- 7 ☒ Gene Cluster Blast Comparative Analysis

- 8 ☐ Whole-genome BLAST results in EMBL output

- 8 ☒ Whole genome PFAM results in EMBL output

-

Fig. 3. antiSMASH front end. The figure shows different sections of the input screen including (1) buttons to load a sample input file or open an example output file; (2) optional input for an e-mail address, where results can be sent; (3) input file options for GBK/EMBL/FASTA files and genome accession numbers; (4) a check box for fungal or other eukaryotic inputs; (5) a check box for gene cluster type selection; (6) a check box for smCOG analysis; (7) a check box for BLAST analysis; and (8) check boxes for CLUSEAN-derived analyses.

8. The files will now be processed automatically and the outputs will be copied to a separate folder “batch_output” in case of directory input, and into the same folder in case of a multiple entry-file input.
9. Complete and integrated multiple-entry input functionality is planned for antiSMASH 2.0, which is currently under development.

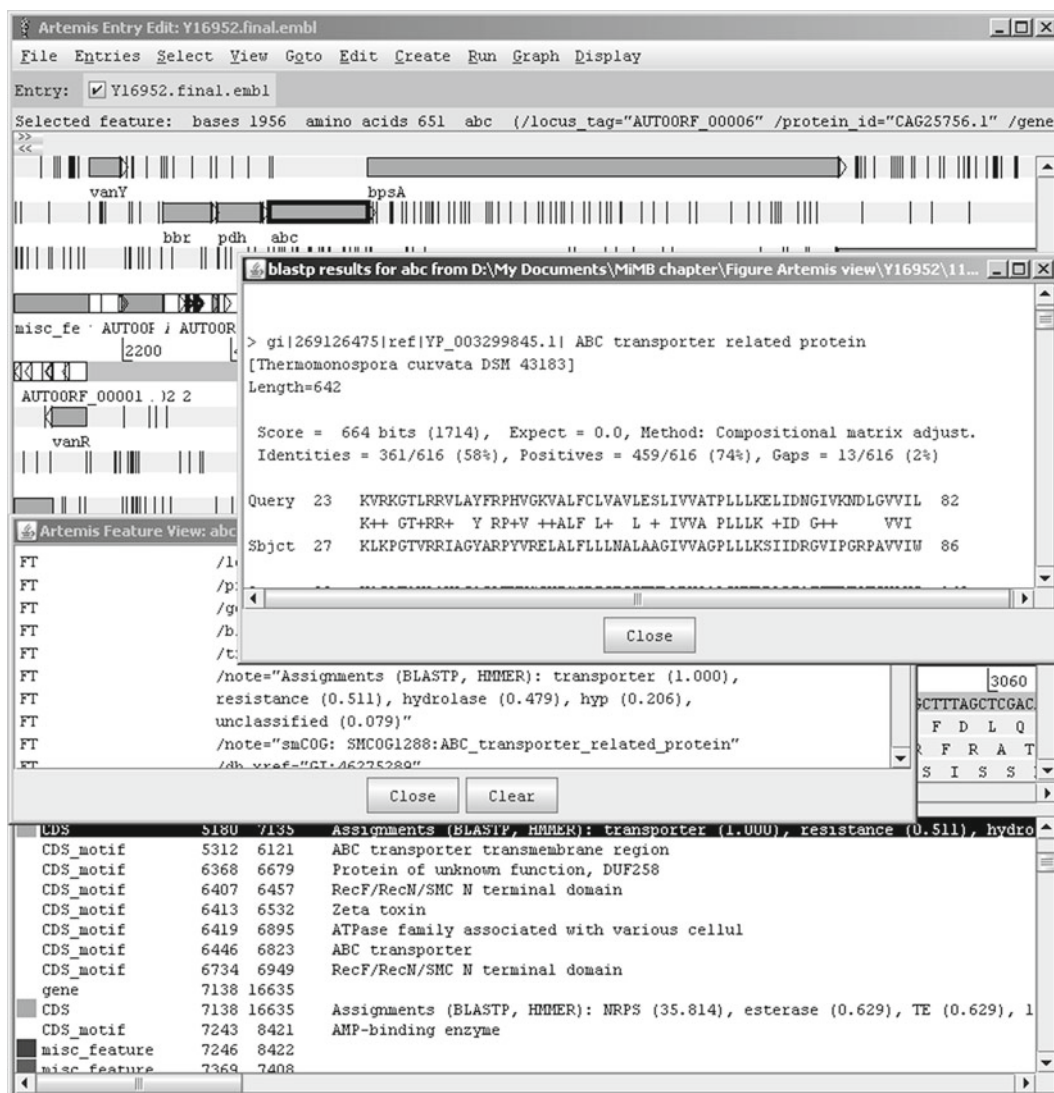


Fig. 4. Screenshot of an antiSMASH EMBL output file viewed in the Artemis genome browser. antiSMASH results such as gene clusters, NRPS/PKS domains, and conserved motifs are displayed as sequence features. Gene function predictions are also available for each gene.

3.2.5. Interpreting antiSMASH Results

antiSMASH has been shown to predict SM clusters with reasonable accuracy (see Notes 3–5) (4). However, the sensitivity and specificity of predictions vary depending on the nature of the data and the type of analyses performed. For example, in order to minimize the number of missed SM genes, it predicts cluster boundaries in a very greedy manner. The user can then modify the predictions based on smCOG and comparative BLAST results (see Note 2). antiSMASH also tends to merge adjacent putative clusters into one gene cluster if they are located within 20 kb (see Note 4). Similarly, products' core structure predictions based on the NRPS

or PKS domain content should be treated with caution, because the backbone molecule is likely to be circularized and/or further modified by decorating enzymes (see Notes 1 and 2).

3.3. Secondary Metabolite Unknown Regions Finder

The SMURF algorithm predicts clustered SM genes in fungal genomic sequences based on genomic context and domain content (3). It identifies putative clusters involved in biosynthesis of polyketides, nonribosomal peptides, indole alkaloids, and their chimeras (see Notes 1 and 2). SMURF relies on the same algorithm as antiSMASH to find backbone enzymes, but uses a different principle to predict cluster boundaries (see Note 2). Both input and output file formats are also distinct from the ones used in antiSMASH. SMURF is freely available at <http://www.jcvi.org/smurf>; it requires user registration and an e-mail address.

3.3.1. Obtaining and Preparing Input Files

To find putative gene clusters in a given fungal genome, SMURF needs two files: (1) a protein FASTA file that contains all protein sequences from the genome and (2) a gene coordinate file in tab-delimited TXT format that contains genomic coordinates for all protein-coding genes (Fig. 1). SMURF does not require a completely sequenced genome, so it can be run on a subset of chromosomes or scaffolds, as long as both files have the same sequence IDs. The tool also allows users to retrieve precomputed clusters for 27 fungal genomes listed on the SMURF Web site. For all other organisms, the user must prepare the two input files as described below. To prepare the first input file for SMURF, the user needs to retrieve all protein sequences for the genome of interest in FASTA format. One way to obtain these sequences is to query the NCBI Entrez general retrieval system (<http://www.ncbi.nlm.nih.gov/entrez>). To limit the query to nonredundant protein sequences, the user needs to use the NCBI Reference Sequence database (RefSeq) by selecting “Protein” from the pull-down menu and entering a query as follows: “*Aspergillus fumigatus*[orgn] AND srcdb refseq[properties]”. Alternatively, he/she can use the NCBI Whole Genome Sequences (WGS) sequencing project database available at <http://www.ncbi.nlm.nih.gov/projects/WGS/WGSprojectlist.cgi>. As with antiSMASH, it is best to use the most current genome build if possible. Protein sequences can be retrieved by clicking first on a species name and then on the Protein link in the Entrez records table. Another alternative would be to use fungal specific sequence data warehouses such as FungiDB (www.fungidb.org), the Broad’s Genome Sequencing Platform (<http://www.broadinstitute.org>), or the Comparative Fungal Genomics Platform (CFGP; <http://cfgp.snu.ac.kr>).

Although many scientific journals require sequence data submission to NCBI prior to publication, some sequenced genomes may not yet be available at the NCBI Web site. For these fungal species, preliminary sequence data may be available directly from

sequencing centers such as the Broad Institute, Joint Genome Institute (JGI), Beijing Genomics Institute (BGI), Wellcome Trust Sanger Institute, or J. Craig Venter Institute (JCVI; former TIGR). To avoid potential data-ownership conflicts, it is recommended to first obtain permission to use the data. If the sequence data is generated by the user or obtained from a sequencing center, the files' format may deviate from the standards. The format can be adjusted using the FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit) or similar tools for formatting sequence data.

To prepare the second input file for SMURF, the user needs to generate a tab-delimited TXT file that has the following elements (values) in each line, separated by tabs: (1) protein ID; (2) chromosome or scaffold ID; (3) gene start coordinate; (4) gene stop coordinate; and (5) protein description if available. The protein IDs must be nonredundant in the dataset. The gene start and end coordinates correspond to the starting (5' end) and ending (3' end) position of each gene on the chromosome/scaffold. The last column is optional and can include the protein name, functional description, or any other functional information.

The data necessary to create the second input file can be obtained from the NCBI ftp site (ftp.ncbi.nih.gov/gene/DATA/ASN_BINARY/Fungi/All_Fungi.ags.gz). The downloaded files can be converted to xml and parsed using the gene2xml tool from NCBI. The tool is available at ftp.ncbi.nih.gov/toolbox/ncbi_tools/cmdline/. If not available at NCBI, the coordinates' information can sometimes be obtained from sequencing center Web sites. The coordinate data are often stored in GFF format, which can be parsed using standard scripts or tools such as the text manipulation option in Galaxy (<http://galaxy.psu.edu>).

3.3.2. Running SMURF

Once the two input files are obtained, they need to undergo a quality control test. FASTA files obtained from non-NCBI sources sometimes need to be "cleaned" by reformatting the file and/or by removing extra characters from the header or the sequences themselves. The coordinates file format needs to be validated manually by checking the order in which values are presented and the delimiter used. The user also needs to make sure that the same protein IDs are used in both the protein FASTA file and the gene coordinate file and that no duplicates are present.

Once both the protein FASTA file and the gene coordinates file are confirmed to be in the required file format, they can be uploaded on the SMURF Web site (Figs. 5 and 6). This can be done by simply copying and pasting the content of the files in the text boxes or by clicking on the "Browse" button and selecting the files from the respective folder. SMURF output files with results are sent to the user by e-mail.

SMURF Input Screen

UPLOAD SEQUENCES

Two types of files are required

- MultiFASTA file containing protein sequences. Headers must contain proteinID as the first element. [View MultiFASTA file example](#)
- Tab delimited TXT file containing the following elements in the order shown below: [View gene coordinates file example](#)

1. protein ID
2. chromosome/contig
3. 5' gene start
4. 3' gene stop
5. protein name/function/definition (if available)

Upload a protein FASTA file No file chosen

Or, Paste protein FASTA

Upload a coordinates file No file chosen

Or, Paste coordinates

If you would like to use processed input files instead, select one of the following genomes that are available on the server

▼

Fig. 5. SMURF front end. The figure shows different sections of the input screen including (1) links to view sample input multiFASTA and coordinate files; (2) a button to upload a protein FASTA file; (3) a text box to paste a protein FASTA file; (4) a text box to paste coordinate files; (5) a Submit button to run SMURF; and (6) a drop-down menu to choose and download precomputed results for 27 fungal genomes.

3.3.3. Interpreting SMURF Results

For each genome, SMURF generates two files: one including the list of putative backbone enzymes and another including the list of putative SM clusters. These files can be used to estimate the number of backbone genes and gene clusters in the genome. The two numbers are not the same, because many backbone genes do not appear to be part of any gene cluster. As with antiSMASH results, caution must be taken when interpreting gene cluster boundaries predicted by SMURF (see Note 2). The accuracy of the predictions can be validated by using gene expression profiling data, if available. Nonetheless, SMURF generates a good preliminary view

SMURF Output Files

File 1: Secondary Metabolism Gene Clusters

Cluster:1									
Backbone_gene_id	Gene_id	Gene_positions	Chromosome-Contig	Gene_order	5'end	3'end	Gene_distance	Domain_score	Annotated_gene_function
AFUA_1G01010	AFUA_1G01010	0 98	67 352044	358955	0 1				"polyketide synthase, putative"
AFUA_1G01010	AFUA_1G01000	1 98	66 351003	349732	1041 1				"oxidoreductase, 2OG-Fe(II) oxygenase family"
AFUA_1G01010	AFUA_1G00990	2 98	65 348512	347632	1220 1				short chain dehydrogenase/reductase family protein
AFUA_1G01010	AFUA_1G00980	3 98	64 346714	344953	918 1				"FAD-dependent oxidase, putative"
AFUA_1G01010	AFUA_1G00970	4 98	63 344153	342453	800 1				"MFS monocarboxylate transporter, putative"

Cluster:2									
Backbone_gene_id	Gene_id	Gene_positions	Chromosome-Contig	Gene_order	5'end	3'end	Gene_distance	Domain_score	Annotated_gene_function
AFUA_1G10380	AFUA_1G10380	0 95	162 447716	428528	0 0				nonribosomal peptide synthase Pes1
AFUA_1G10380	AFUA_1G10370	1 95	161 426122	428041	487 1				"MFS multidrug transporter, putative"
AFUA_1G10380	AFUA_1G10360	2 95	160 422521	425058	1064 0				conserved hypothetical protein
AFUA_1G10380	AFUA_1G10355	3 95	159 421515	421046	1006 1				"26 proteasome complex subunit Sem1, putative"
AFUA_1G10380	AFUA_1G10350	4 95	158 419083	420498	548 0				"phosphoglycerate kinase PgkA, putative"
AFUA_1G10380	AFUA_1G10340	5 95	157 417419	418337	746 0				"integral membrane protein, Mpv17/PMP22 family, putative"
AFUA_1G10380	AFUA_1G10330	6 95	156 414576	416503	916 0				conserved hypothetical protein
AFUA_1G10380	AFUA_1G10320	7 95	155 414256	411705	320 0				"formin binding protein (FNB3), putative"
AFUA_1G10380	AFUA_1G10310	8 95	154 408473	410949	756 1				"RNase L inhibitor of the ABC superfamily, putative"

File 2: Backbone genes

Backbone_gene_id	Annotated_gene_function	Chromosome-Contig	Gene_order	5'_end	3'_end	SMURF_backbone_gene_prediction
AFUA_1G01010	"polyketide synthase, putative"	98 67 352044	358955		PKS	
AFUA_1G10380	nonribosomal peptide synthase Pes1	95 162 447716	428528		NRPS	
AFUA_1G17200	nonribosomal siderophore peptide synthase SidC	95 843 2441629	2455970		NRPS	
AFUA_1G17740	"polyketide synthase, putative"	95 894 2655126	2663085		PKS	
AFUA_2G01290	"polyketide synthase, putative"	57 120 314706	306145		PKS	
AFUA_2G05760	"beta-ketoacyl synthase (Cem1), putative"	57 553 1621987	1623626		PKS-Like	
AFUA_2G17600	condidial pigment polyketide synthase PksP/Alb1	92 967 2817817	2824478		PKS	
AFUA_2G17990	dimethylallyl tryptophan synthase FgaPT1	92 1005 2913852	2912488		DMAT	
AFUA_2G18040	dimethylallyl tryptophan synthase FgaPT2	92 1010 2925193	2923679		DMAT	
AFUA_3G01410	"polyketide synthase, putative"	100 126 360985	352856		PKS	
AFUA_3G02530	"PKS-like enzyme, putative"	100 230 634289	632049		PKS-Like	
AFUA_3G02570	"polyketide synthase, putative"	100 234 653701	646906		PKS	
AFUA_3G02670	"NRPS-like enzyme, putative"	100 244 694320	697670		NRPS-Like	
AFUA_3G03350	nonribosomal peptide synthase SidE	100 307 892438	898767		NRPS	
AFUA_3G03420	nonribosomal peptide synthase SidD	100 313 908168	914474		NRPS	
AFUA_3G12920	"nonribosomal peptide synthase GliP-like, putative"	93 238 648974	642245		NRPS	
AFUA_3G12930	"dimethylallyl tryptophan synthase SirD-like, putative"	93 237 641551	640080		DMAT	
AFUA_3G13730	"nonribosomal peptide synthase, putative"	93 159 459847	455975		NRPS	
AFUA_3G14700	"polyketide synthase, putative"	93 64 181917	174183		PKS	
AFUA_3G15270	"nonribosomal peptide synthase, putative"	93 11 68646 61531	NRPS			
AFUA_4G00210	"polyketide synthase, putative"	109 134 384731	390228		PKS	
AFUA_4G14560	"polyketide synthase, putative"	96 30 81245 86775	PKS			
AFUA_5G10120	"NRPS-like enzyme, putative"	94 473 1345356	1341532		NRPS-Like	
AFUA_5G12730	"nonribosomal peptide synthase, putative"	94 225 633905	608358		NRPS	
AFUA_6G03480	"NRPS-like enzyme, putative"	101 186 545229	540113		NRPS-Like	
AFUA_6G08560	"NRPS-like enzyme, putative"	108 225 704656	707884		NRPS-Like	
AFUA_6G09610	"nonribosomal peptide synthase, putative"	108 327 1024733	1020915		NRPS	
AFUA_6G09660	nonribosomal peptide synthase GliP	108 332 1040501	1033997		NRPS	
AFUA_6G12050	"nonribosomal peptide synthase, putative"	108 556 1694970	1698884		NRPS	
AFUA_6G12080	"nonribosomal peptide synthase, putative"	108 559 1716682	1704693		NRPS	
AFUA_6G13930	"polyketide synthase LovB-like, putative"	108 737 2239872	2232481		PKS	

Fig. 6. SMURF output. Typical SMURF results are presented: a TXT file with gene clusters and a TXT file with backbone genes and their functional descriptions.

of the species' secondary metabolome. If desired, further predictions can be made of the substrate specificity and core structure based on domain architecture of the backbone enzymes such as NRPSs and PKSs using other tools described in this chapter. To visualize the genomic context of the identified clusters, the user can use available browsers such as the CFGP (<http://cfgp.snu.ac.kr>) or FungiDB (<http://www.fungidb.org>). Batch processing of multiple genomes is currently not available in SMURF.

3.3.4. SMURF Algorithm

To better interpret SMURF output, it may be helpful to understand the underlying algorithm. At the heart of SMURF lie hidden

Markov models (HMM) for each PFAM and TIGRFAM domain that have been associated with backbone and decorating proteins (16, 17). The HMMER program (<http://hmmer.janelia.org>) is employed to search for these domains in protein dataset (16). NRPS enzymes are identified as enzymes with at least one module composed of an amino acid adenylation domain (A), a thiolation domain (PCP), and a condensation domain (C). PKS enzymes are identified as enzymes with at least one acyl transferase domain (AT), a beta-ketoacyl synthase C-terminal domain (BKS-C), and a beta-ketoacyl synthase N-terminal domain (BKS-N). Hybrid PKS–NRPS enzymes are identified as enzymes with at least one instance from each set of three domains listed above.

NRPS-like enzymes are identified with a combination of at least two domains from any of those in the NRPS enzyme module; a combination of an A domain and a NAD_binding_4 domain; or a combination of an A domain and short chain dehydrogenase domain. PKS-like enzymes were identified with a combination of at least two domains from any of those in the PKS enzyme module. To eliminate false positives among PKS-like enzymes, they were defined as proteins with AT, BKS-C, and BKS-N domains that scored below a trusted HMM cutoff. To identify false positives such as alpha-aminoadipate reductase among NRPSs, candidate NRPSs are screened for the presence of the C-terminal domain of L-aminoadipate-semialdehyde dehydrogenase alpha subunit with the cutoff score set above the default cutoff. Those that score below the cutoff are removed from the final list of putative NRPSs. Prenyltransferase enzymes are identified as enzymes with at least one DMATS-type prenyltransferase domain (DMATS). The corresponding de novo HMM model for this domain (TIGR03429) was created in this study from the seed alignment generated using *A. fumigatus* dimethylallyl tryptophan synthase FtmPT2 as a seed sequence as previously described (16).

Putative decorating enzymes are predicted by checking for the presence of one of the 27 SM-defining domains, as well as by their genomic neighborhood, i.e., the distance to the nearest backbone genes. The algorithm first identifies SM-defining domains in the window of ± 20 genes on each side of a backbone gene. The algorithm uses two fixed parameters: d and y to define the cluster's boundaries. d is the maximum intergenic distance permitted between two adjacent genes in the same cluster, while y is the maximum number of SM domain-negative genes. These parameters have been set as $d = 3,814$ bp and $y = 10$ genes, based on optimization with experimental data. Finally, additional genes are trimmed at both ends of the cluster until the algorithm reaches the first gene that contains an SM-defining domain. If SMURF predicts overlapping clusters, they are merged into one (see Note 4).

3.4. NRPSPredictor

NRPSPredictor ((9, 10) is a widely used tool for prediction of substrate specificities of NRPS enzymes. Predictions are based on the sequence of the adenylation (A) domain, which is responsible for the recruitment of the amino acid monomers by NRPS enzymes. NRPSPredictor is a Web-based tool, available at <http://nrps.informatik.uni-tuebingen.de>. A new version, NRPSPredictor2, based on updated training data, has recently been published (10). Unlike the original version, NRPSPredictor2 contains a separate fungal module. The input consists either of a full NRPS protein sequence in FASTA format or of a set of extracted A domain signatures based on amino acid residues surrounding the A domain active site (Table 1). Input sequences can be pasted in the text box called “Sequence to analyze”) or uploaded using the “Browse” button. The output contains (1) a list of A domains identified in the submitted NRPS sequences, (2) their predicted substrate specificities, (3) support-vector machine scores for domains, and (4) substrate specificity of their nearest homolog in the NRPSPredictor signature database.

The tool relies on two algorithms: (1) a machine-learning technique called transductive support-vector machines to predict the substrate specificity of an unknown sequence and (2) a sequence similarity search using amino acid residues surrounding the A domain active site (active site code) as a query against a set of codes from experimentally characterized A domains. It also provides support-vector machine predictions at four hierarchical levels, from predictions of gross physicochemical properties of the amino acid to predictions of single amino acid specificity. If substrate specificity cannot be predicted based on training data, this module can still predict the general nature of the substrate (e.g., hydrophilic or hydrophobic). Since antiSMASH includes the NRPSPredictor2 module, this pipeline can also be used to predict substrate specificities of NRPS enzymes. Yet, for a quick analysis of single NRPS genes, the NRPSPredictor2 Web server is most efficient.

3.5. CLUster SEquence ANalyzer

CLUSEAN (5) is a comprehensive SM analysis pipeline for bacterial genomes. The latest version has been adapted to also process fungal genomes, and has been merged into the antiSMASH pipeline (represented by part of the NRPS/PKS domain detection module, as well as the genome-wide PFAM and BLAST options) (4). A stand-alone version of CLUSEAN is available at <http://redmine.secondarymetabolites.org/projects/clusean>. It is an open-source stand-alone tool, which requires installation. The preferred environment is Linux or Unix, although it is compatible with MS Win 2000, XP, and Vista. Installation of additional tools is also required, including NCBI BLAST, HMMer, SVMlight, NRPSPredictor (described above), Artemis, and Perl 5.8 or higher. CLUSEAN takes DNA sequences in EMBL format as input and generates functional predictions for NRPS and type I PKS genes as well as substrate specificity predictions for NRPS genes.

3.6. Structure Based Sequence Analysis of Polyketide Synthases

Designed mainly for analysis of bacterial type I modular PKS genes, the SBSPKS Web server offers some interesting features, which complement those offered in SMURF and antiSMASH (7). SBSPKS input consists of protein sequences in FASTA format, to a maximum of ten sequences (Table 1). Input sequences can be entered in a text box manually by copying and pasting. In addition to domain architecture and docking analysis (also present in antiSMASH), SBSPKS hosts a database of PKS genes associated with known compounds. This feature can be used to link PKS domain architecture with the polyketide structure and to generate a structural homology model of the entire PKS module. The model can be used to view putative molecular interactions in 3D as well as to make inferences on the probable functions of highly conserved residues. SBSPKS can be accessed at <http://www.nii.ac.in/sbspks.html>.

3.7. Natural Product searcher

NP.searcher (8) is a tool for detecting genes encoding PKSs, NRPSs, and some terpene synthases in bacterial and fungal genomes. Although a recent benchmark (4) showed that the tool has limited sensitivity, it has the interesting functionality to generate putative chemical structures for polyketide or nonribosomal peptide compounds, including circularization and glycosylation variants. It is a Web-based tool that takes DNA sequences in FASTA format as input (Table 1). Input sequences can be pasted in a text box or uploaded using the “Browse” button. The output contains chromosomal coordinates for putative NRPS and PKS genes and the number of detected PKS and NRPS modules. NP.searcher is available at <http://dna.sherman.lsi.umich.edu>.

3.8. ClustScan

One of the earliest programs for detection of SM gene clusters is ClustScan (6), a commercial tool that runs on an external server using a stand-alone client program that can be downloaded from their Web site. As input, the program takes DNA sequences in FASTA format or any other format supported by ReadSeq, a biosequence conversion tool (<http://www.bimas.cit.nih.gov/molbio/readseq/>). Output includes annotated backbone genes and domain prediction. Interestingly, ClustScan can also use custom-built HMM to identify SM genes in bacterial genomes (see Note 3). The program output is interactive, which allows users to edit annotations based on their preference. Recently, an algorithm for detection of recombination events between gene clusters has also been added to the tool (18). In order to use ClustScan, registration is required, although the program can be used free of charge on a trial basis for 30 days. It is available from <http://bioserv.pbf.hr/cms/>.

4. Notes

1. Necessity of experimental confirmation of gene cluster prediction validity

Further genetic and structural elucidation studies are needed to experimentally confirm putative gene clusters detected by these tools, and to characterize the structures of the compounds that they encode. Although some tools, such as antiSMASH and NP.searcher, can give an approximation of the core structure of some classes of compounds, the circularization and tailoring steps remain difficult to predict.

2. Specificity and sensitivity of gene cluster boundary predictions

Both antiSMASH and SMURF predict genes encoding backbone enzymes with almost 100% accuracy. However, other genes in a cluster are more difficult to predict. Some of them contain unknown domains, which also contribute to poor predictions. Therefore, predicting the exact gene cluster boundaries remains a challenge, which antiSMASH and SMURF currently solve in a somewhat different manner. SMURF utilizes a list of 27 SM-specific domains to try and give an exact prediction of the gene cluster boundaries. antiSMASH, on the other hand, uses a safe margin of 5, 10, 15, or 20 kb around the backbone genes, depending on the gene cluster type. The user can then choose cluster boundaries based on additional output from smCOG gene family predictions and cluster-cluster alignments with related gene clusters. By design, both tools tend to overpredict cluster boundaries. SMURF, for example, over-predicted the number of genes included in each cluster by 8.4 on average (3). Presently, further tool refinement is limited by the number of experimentally characterized clusters and related transcriptomics, proteomics, and metabolomics datasets.

3. Problems associated with genome annotation

Some of the tools described here (notably, ClustScan and antiSMASH) can annotate a DNA sequence using gene predictors, Glimmer, GlimmerHMM, or Genemark. While bacterial genome annotation tools may work reasonably well, gene predictors mis-annotate a large portion of genes in eukaryotic genomes, including some model organism genomes (19). This is especially true for fungal genomes, which often have small introns, tightly spaced genes, and other unusual features (20). Furthermore, fungal NRPS and PKS genes are extremely long and tend to be fragmented in poorly annotated genomes. This in turn may cause problems in domain identification and subsequent predictions. To obtain more reliable gene predictions, it is best to use consensus or post-processing methods (21).

4. Artificially merged and split gene clusters

Experimental validation is necessary to determine whether all these genes are indeed part of one “supercluster” with multiple backbone genes, or whether two or more separate gene clusters happen to be located very closely together. Close inspection of the “homologous gene clusters” section in antiSMASH may aid in obtaining a better prediction in such case. If close homologs of the supercluster include multiple separate gene clusters, most likely the supercluster needs to be split into two or more clusters.

5. One pathway represented by two or more subclusters

Some pathways may be encoded by two or more subclusters found at different locations on chromosomes (e.g., (22)). These types of clusters are more difficult to predict, because some of them may lack a backbone gene and most programs rely on identification of backbone genes to “anchor” clusters. In some cases, manual examination of the “homologous gene clusters” section in antiSMASH may be helpful.

Acknowledgements

We thank Suman Pakala and Bill Nierman at JCVI for critical suggestions and comments. The work of MHM was supported by the Dutch Technology Foundation, which is the applied-science division of The Netherlands Organisation for Scientific Research and the Technology Programme of the Ministry of Economic Affairs under STW grant number 10463. This project has been funded in part with federal funds to NDF from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under contract numbers N01-AI-30071 and HHSN272200900007C.

References

1. Winter JM, Behnken S, Hertweck C (2011) Genomics-inspired discovery of natural products. *Curr Opin Chem Biol* 15(1):22–31
2. Keller NP, Hohn TM (1997) Metabolic pathway gene clusters in filamentous fungi. *Fungal Genet Biol* 21(1):17–29
3. Khaldi N, Seifuddin FT, Turner G, Haft D, Nierman WC, Wolfe KH, Fedorova ND (2010) SMURF: Genomic mapping of fungal secondary metabolite clusters. *Fungal Genet Biol* 47(9):736–741
4. Medema MH, Blin K, Cimermanic P, de Jager V, Zakrzewski P, Fischbach MA, Weber T, Takano E, Breitling R (2011) antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. *Nucleic Acids Res* 39:W339–W346
5. Weber T, Rausch C, Lopez P, Hoof I, Gaykova V, Huson DH, Wohlleben W (2009) CLUSEAN: a computer-based framework for the automated analysis of bacterial secondary metabolite biosynthetic gene clusters. *J Biotechnol* 140(1–2):13–17
6. Starcevic A, Zucko J, Simunkovic J, Long PF, Cullum J, Hranueli D (2008) ClustScan: an integrated program package for the semi-automatic annotation of modular biosynthetic gene

- clusters and in silico prediction of novel chemical structures. *Nucleic Acids Res* 36(21): 6882–6892
7. Anand S, Prasad MV, Yadav G, Kumar N, Shehara J, Ansari MZ, Mohanty D (2010) SBSPKS: structure based sequence analysis of polyketide synthases. *Nucleic Acids Res* 38:W487–W496
 8. Li MH, Ung PM, Zajkowski J, Garneau-Tsodikova S, Sherman DH (2009) Automated genome mining for natural products. *BMC Bioinformatics* 10:185
 9. Rausch C, Weber T, Kohlbacher O, Wohlleben W, Huson DH (2005) Specificity prediction of adenylation domains in nonribosomal peptide synthetases (NRPS) using transductive support vector machines (TSVMs). *Nucleic Acids Res* 33(18):5799–5808
 10. Röttig M, Medema MH, Blin K, Weber T, Rausch C, Kohlbacher O (2011) NRPSpredictor2—a web server for predicting NRPS adenylation domain specificity. *Nucleic Acids Res* 39(2):W362–W367
 11. Lansini G, Demain AL (1999) Biology of the prokaryotes. Georg Thieme, Stuttgart
 12. Majoros WH, Pertea M, Salzberg SL (2004) TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics* 20(16):2878–2879
 13. Delcher AL, Bratke KA, Powers EC, Salzberg SL (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics* 23(6):673–679
 14. Cantarel BL, Korf I, Robb SM, Parra G, Ross E, Moore B, Holt C, Sanchez Alvarado A, Yandell M (2008) MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Res* 18(1):188–196
 15. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M et al (2008) The RAST Server: rapid annotations using subsystems technology. *BMC Genomics* 9:75
 16. Sonnhammer EL, Eddy SR, Birney E, Bateman A, Durbin R (1998) Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res* 26(1): 320–322
 17. Haft DH, Selengut JD, White O (2003) The TIGRFAMs database of protein families. *Nucleic Acids Res* 31(1):371–373
 18. Starcevic A, Diminic J, Zucko J, Elbekali M, Schlosser T, Lisfi M, Vukelic A, Long PF, Hranueli D, Cullum J (2011) A novel docking domain interface model predicting recombination between homoeologous modular biosynthetic gene clusters. *J Ind Microbiol Biotechnol* 38(9):1295–1304. doi:10.1007/s10295-10010-10909-10290
 19. Wortman JR, Gilsenan JM, Joardar V, Deegan J, Clutterbuck J, Andersen MR, Archer D, Bencina M, Braus G, Coutinho P et al (2009) The 2008 update of the *Aspergillus nidulans* genome annotation: a community effort. *Fungal Genet Biol* 46(Suppl 1):S2–S13
 20. Ma L-J, Fedorova ND (2010) A practical guide to fungal genome projects. *Mycol Int J Fungal Biol* 1(1):9–24
 21. Pati A, Ivanova NN, Mikhailova N, Ovchinnikova G, Hooper SD, Lykidis A, Kyrpides NC (2010) GenePRIMP: a gene prediction improvement pipeline for prokaryotic genomes. *Nat Methods* 7(6):455–457
 22. Nicholson MJ, Koulman A, Monahan BJ, Pritchard BL, Payne GA, Scott B (2009) Identification of two aflatoxin biosynthesis gene loci in *Aspergillus flavus* and metabolic engineering of *Penicillium paxilli* to elucidate their function. *Appl Environ Microbiol* 75(23): 7469–7481



<http://www.springer.com/978-1-62703-121-9>

Fungal Secondary Metabolism

Methods and Protocols

Keller, N.P.; Turner, G. (Eds.)

2012, XII, 288 p., Hardcover

ISBN: 978-1-62703-121-9

A product of Humana Press